REMARKS

Applicant respectfully traverses and requests reconsideration.

Claims 1-4 and 7-8 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent Publication 20040103272 to Zimmer et al. ("Zimmer") in view of U.S. Patent Publication 20040186988 to Polyudov ("Polyudov"). Claim 5 stands rejected under 35 U.S.C. § 103(a) as being unpatentable over Zimmer in view of Polyudov and U.S. Patent No. 4,922,451 to Lo et al. ("Lo"). Claim 6 stands rejected under 35 U.S.C. § 103(a) as being unpatentable over Zimmer in view of Polyudov and U.S. Publication 20030005314 to Gammel et al. ("Gammel"). Claims 9-13 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Polyudov in view of Zimmer. Claims 14-20 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Zimmer in view of Polyudov and Gammel. Claim 21 stands rejected under 35 U.S.C. § 103(a) as being unpatentable over Polyudov and Zimmer.

Zimmer is directed to using a processor cache as RAM during a platform initialization. (*See* Title). The reference appears to teach a system having a single processor having an execution core and one or more caches. (FIG. 1, elements 12-18). The processor may be coupled to an interface that allows communication with system memory 25 and system ROM 20. (FIG. 2). During the initialization process, early initialization firmware may "run out" of the system ROM 20 and "the initial contents of the initialization process, prior to the availability of system memory 25, may be stored in the caches 16 and 18 on the processor 10. The caches 16 and 18 may act as static random access memory …." (¶ 0012). Data may be locked in the instruction cache 16 and/or the data cache 18. (¶¶ 0013, 0015). The contents of the cache remain valid after locking. (¶ 0014). Thus, according to Zimmer, the data cache 18 is functioning as a cache-as-RAM. This corresponds to a mode where the data is not evicted from

the cache 18. (¶ 0018). The code and data locked in the caches 16 and 18 may optionally run as an algorithm to authenticate permanent or system memory 25 initialization code (to initialize the system memory). (¶ 0019). Upon initialization of system memory 25, the cache code and data may be copied to system memory. (¶ 0020).

As identified by the Office Action, Zimmer does not disclose a multi-processor environment. (p. 2, ¶ 4). Applicant agrees that Zimmer fails to teach or suggest this environment. In fact Applicant notes that Zimmer appears to be limited to a single processor that reads its own cache to execute code and data locked therein. Zimmer is silent as to any other features.

Polyudov appears to be directed to a system for updating memory devices in a multi-processor computer system. The multi-processor computer system includes two groups of processors 4A-4D and 6A-6D. Each group of processors communicates with a hardware interface called a node that provides access only to flash memory devices associated with that specific node. (¶ 0003, 0021). For example, processors 4A-4D communicate with node 18 to access flash memory devices 24A-24B. Processors 6A-6D communicate with node 20 to access flash memory devices 26A-26B. (FIG. 1). Processors 4A-4D may not access flash memory devices 26A-26B; Processors 6A-6D may not access flash memory devices 24A-24B. (¶ 0021). In order to avoid the manual launch of multiple copies of an update utility on each node used to update the various flash memory devices in a multi-processor platform (¶ 0003), Polyudov discloses an alternate system and method.

The Polyudov system and method utilizes a first processor (e.g., a main or host processor) to inform a second processor (e.g., an identified processor) of the location of the update code 61 so that the second processor can more efficiently update flash memories that the

first processor cannot access. More specifically, a first processor (termed a main or boot processor) from a first group of processors 4A-4D launches an updated utility (stored in an input/output device 54) having an associated update code 61 and copies it to a memory location in RAM 10 accessible by all processors in the system. (¶¶ 0020, 0026, 0028). The main processor (one of 4A-4D) communicates the memory location in the RAM 10 storing the update code 61 to one of the processors (called the identified processor) in the second group of processors 6A-6D using a process called inter-processor interrupt ("IPI"). (¶ 0030). Thereafter, the main processor executes the update code 61 to update the flash memory devices 24A-24B (devices that only it can access). Similarly, the identified processor executes the update code 61 to update the flash memory devices 26A-26D (devices that only it can access). (¶ 0031). Execution of the update code 61 is performed after each processor respectively makes a call to the memory location in the RAM 10 in which the update code 61 is stored. (*Id.*).

Claim 1

The method of claim 1 features executing a memory initialization and sizing operation using the data in the cache memory disposed in a central processing unit using another processor operatively coupled to the central processing unit (e.g., a graphics processing within a common chip). (Emphasis added). The references cited by the Office Action fail to teach or suggest at least this claim feature. For this reason, claim 1 is believed to be allowable.

The Office Action asserts that paragraphs 0019, 0021 and 0023 of Zimmer teach executing a memory initialization and sizing operation using the data in the cache memory and that paragraph 0020 of "Polyudov discloses a method wherein the start-up operation is performed by another processor [main processor] operatively coupled to the central processing unit [other processors]." (p. 2, ¶¶ 3-5). Accordingly, the Office Action asserts that it would have

been obvious to combine the references because: (1) Polyudov teaches that the "main boot processor boots with updated information stored in another processor's cache at another node by communicating directly with the other processor via IPI [inter-processor interrupt] instead of the inaccessible flash memory" (emphasis added); and (2) this would increased processing efficiency by task dividing. (pp. 2-3, ¶ 6). This is improper for at least the reason that it mischaracterizes the teachings of Polydov.

Contrary to the Office Action's citation, Polyudov does not appear to teach that the main processor boots with updated information stored in another processor's cache at another node by communicating directly with the other processor via IPI [inter-processor interrupt] instead of the inaccessible flash memory. This is incorrect. First, the term "cache" is not used in the Polyudov reference. Thus, the above statement is factually inconsistent with the express teachings of the reference. Second, the main processor boots with updated information stored in its own flash memory (that the identified processor cannot access) and the identified processor boots with updated information stored its flash memory (that the main processor cannot access). The information in each flash memory was called by the respective processor that executes it. Neither processor can access the flash memory of the other.

Buttressing this explanation, Applicant notes that the Office Action's assertion that the main processor boots with updated information stored in another processor's cache (or stored in another processor's flash memory) is contradictory to the actual constraints recognized by Polyudov (*See e.g.*, ¶ 0021) and would render the method disclosed in Polyudov trivial and unnecessary. For instance, Applicant notes that the main processor in the Polyudov would not need to use the IPI technique if it could directly communicate with another processor's flash memory in a different processor group.

For the reasons discussed above, Polyudov has been mischaracterized. Accordingly, the Office Action has failed to provide a *prima facie* rejection. For this reason, no combination of the cited prior art teaches or suggests the features of claim 1.

Claims 9, 17 and 21

Claims 9, 17 and 21 contain features similar to those articulated above. For at least the reasons identified above, these claims are also believed to be allowable.

Dependent Claims

Claims 2-8 depend on allowable base claim 1. Claims 10-16 depend on allowable base claim 9. Claims 18-20 depend on allowable base claim 17. The aforementioned dependent claims further contain additional novel, non-obvious and patentable subject matter and are believed to be in proper condition for allowance for at least the reasons stated above.

Accordingly, Applicant respectfully submits that the claims are in condition for allowance and that a timely Notice of Allowance be issued in this case. The Examiner is invited to contact the below-listed attorney if the Examiner believes that a telephone conference will advance the prosecution of this application.

Respectfully submitted,

Date: 3/17/09

By: _____
Christopher J. Reckamp
Registration No. 34,414

Vedder, Price, Kaufman & Kammholz, P.C.
222 N. LaSalle Street
Chicago, IL 60601
(312) 609-7500
FAX: (312) 609-5005